Creating Chatbot with Amazon Lex

What Is Amazon Lex Bot?

Amazon Lex is a fully managed service for building conversational interfaces into any application using voice and text. It provides deep learning functionalities like:

- Automatic Speech Recognition (ASR) for converting speech to text
- Natural Language Understanding (NLU) to recognize the intent of the text

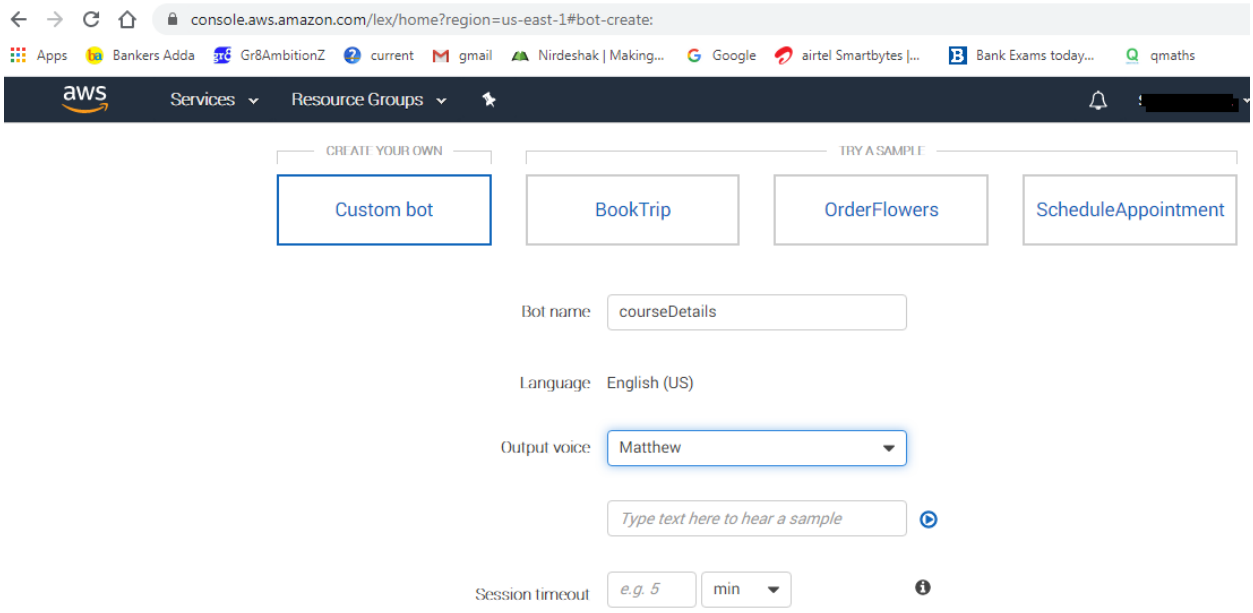
Amazon Lex enables you to quickly & easily build chatbots with highly engaging user experiences and lifelike conversational interactions. Now you might be wondering,

Benefits of using Amazon Lex:

- **Simplicity:** It offers an easy-to-use console to create your own chatbot in minutes & predefined bots to help you get started.
- **Inbuilt Technologies:** You supply just a few example phrases, and Amazon Lex builds a complete natural language model through which the bot can interact using voice and text.
- **Seamless deployment and scaling:** As the user engagement increases, you don't need to worry about provisioning hardware and managing infrastructure to power your bot experience.
- **Built-in integration with AWS:** Amazon Lex allows integrating with many other services on the AWS platform including AWS Lambda, Amazon CloudWatch, Amazon Cognito, and Amazon DynamoDB & many others.
- **Cost-Effective:** With Amazon Lex, there are no upfront costs or minimum fees. You will have to pay only for the text or speech requests that you make

To create the bot

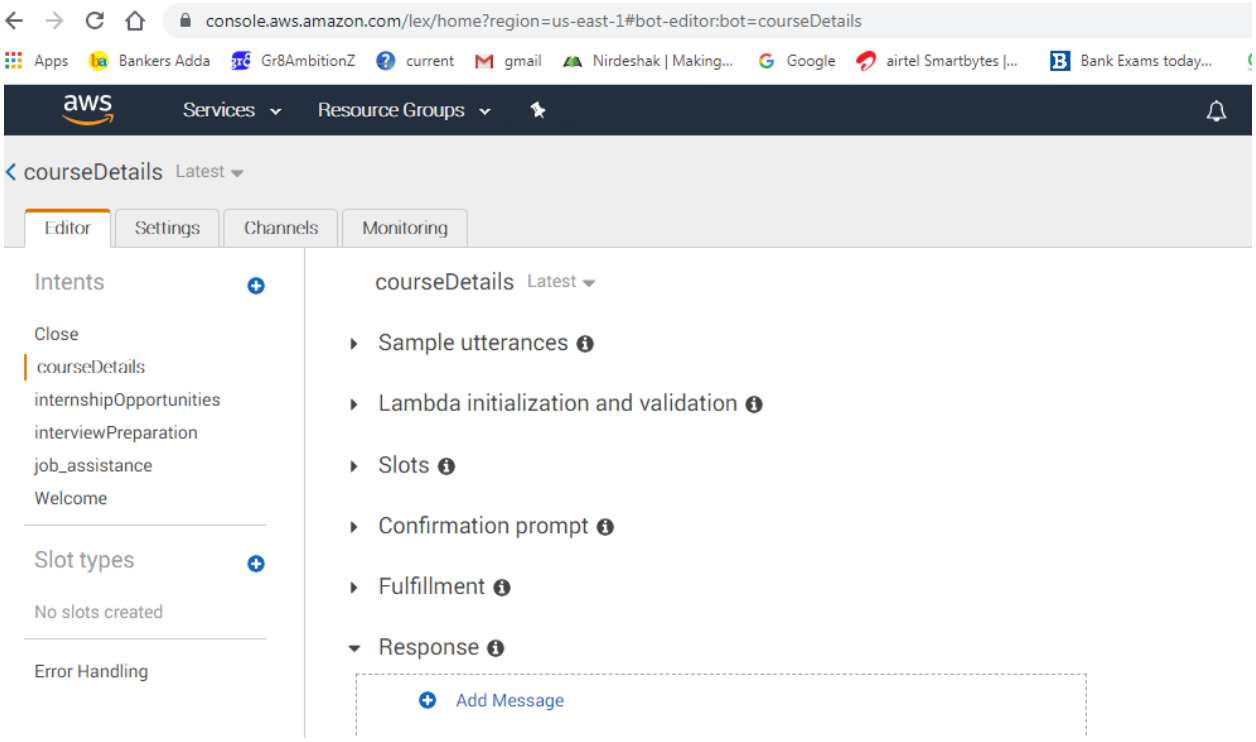
- Sign in to the AWS Management Console and open the Amazon Lex console at https://console.aws.amazon.com/lex/.
- Create a bot.
- To create a bot, choose **Get Started** and then choose **Create**.
- On the **Create your Lex bot** page, choose **Custom bot** and provide the following information:
    - **App name**: courseDetails
    - **Output voice**: Matthew
    - **Session timeout** : 5 minutes.
    - **Child-Directed**: Choose the appropriate response.
- Choose **Create**.



- The console sends Amazon Lex a request to create a new bot. Amazon Lex sets the bot version to $LATEST.

Create an Intent

- **In the Amazon Lex console, choose the plus sign (+) next to** Intents**, and then choose** Create new intent**.**
- **In the** Create intent **dialog box, type the name of the intent , and then choose** Add**.**
- **For courseDeatils create the following intents.**
    - **Welcome**
    - **Job_assistance**
    - **courseDetails**
    - **internshipOpportunities**
    - **interviewPreparation**
    - **close**



Configure the intent

On the **welcome** configuration page, configure the intent as follows:

Sample utterances – Type the following strings. The curly braces {} enclose slot names.

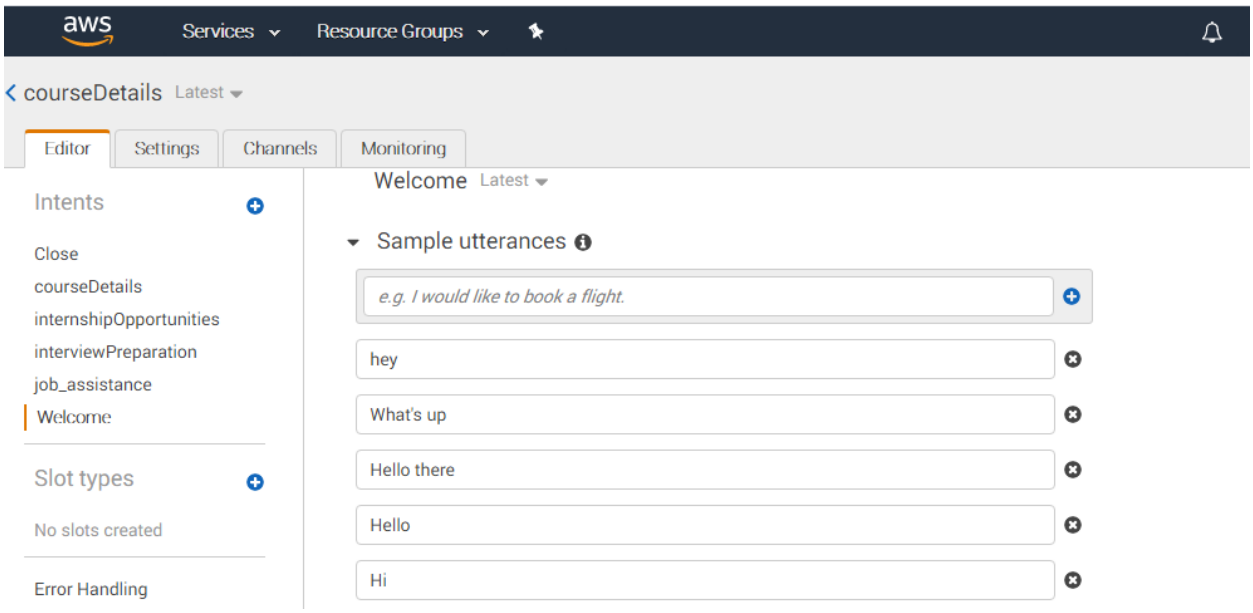
- **Hi**
- Hello
- Hey
- Hello there
- What' up

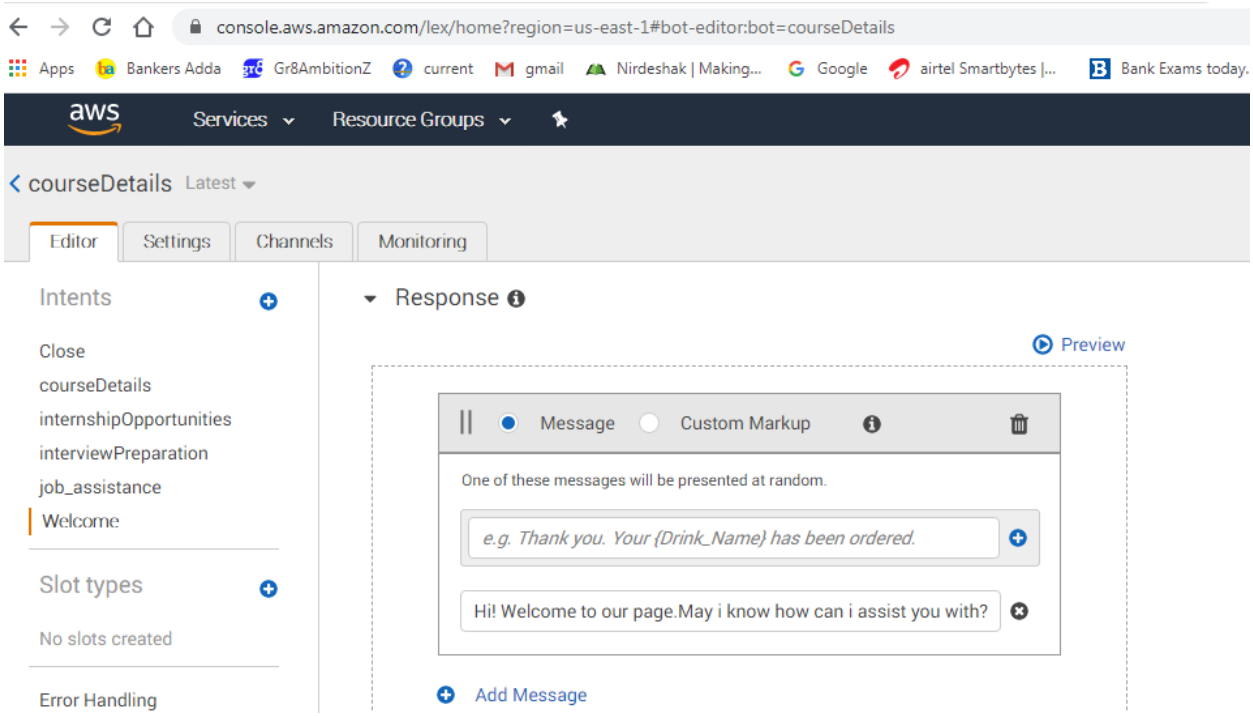
Responses-

- Hi! Welcome to our page.May i know how can i assist you with?

Check the wait for user reply checkbox and add the following in the message box.

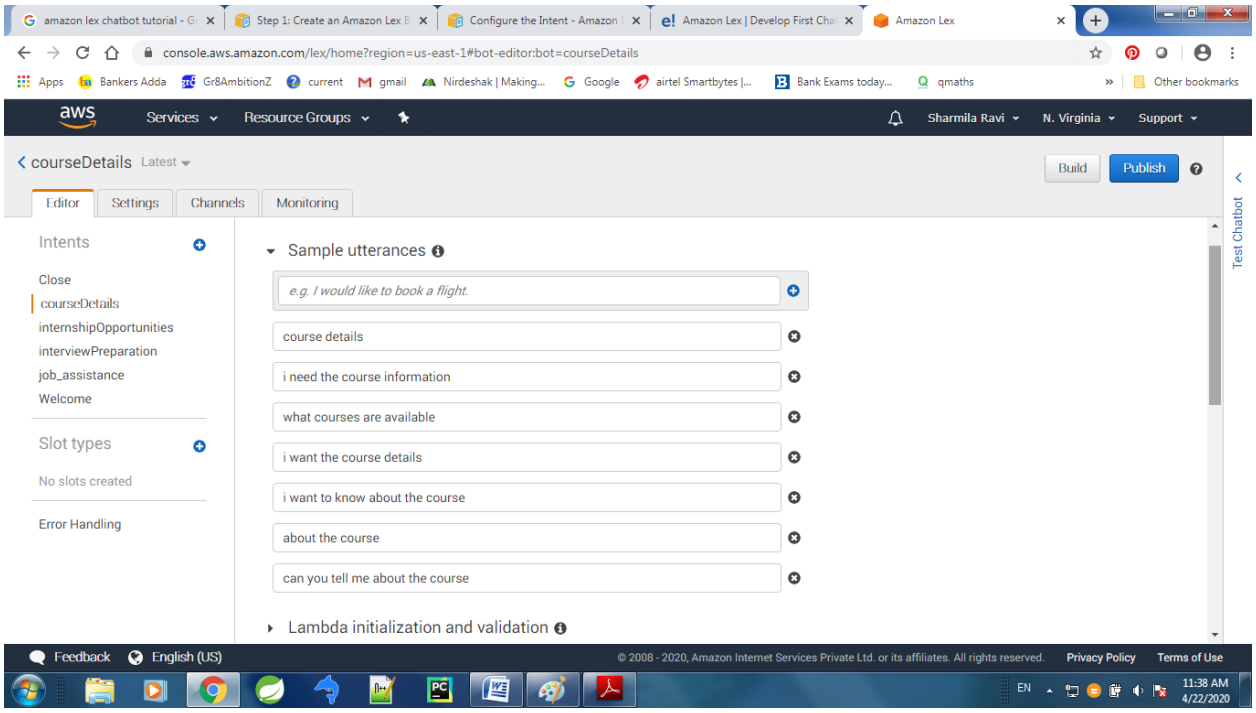
- Okay.Thank you for visiting our page


aws
Services
Resource Groups
courseDetails
Latest
Editor
Settings
Channels
Monitoring
Intents
Close
courseDetails
internshipOpportunities
interviewPreparation
job_assistance
Welcome
Slot types
No slots created
Error Handling
Welcome
Latest
Sample utterances
e.g. I would like to book a flight.
hey
What's up
Hello there
Hello
Hi


console.aws.amazon.com/lex/home?region=us-east-1#bot-editor:bot=courseDetails
Apps
Bankers Adda
Gr8AmbitionZ
current
gmail
Nirdeshak | Making...
Google
airtel Smartbytes |...
Bank Exams today.
aws
Services
Resource Groups
courseDetails
Latest
Editor
Settings
Channels
Monitoring
Intents
Close
courseDetails
internshipOpportunities
interviewPreparation
job_assistance
Welcome
Slot types
No slots created
Error Handling
Response
Preview
Message
Custom Markup
One of these messages will be presented at random.
e.g. Thank you. Your {Drink_Name} has been ordered.
Hi! Welcome to our page.May i know how can i assist you with?
Add Message

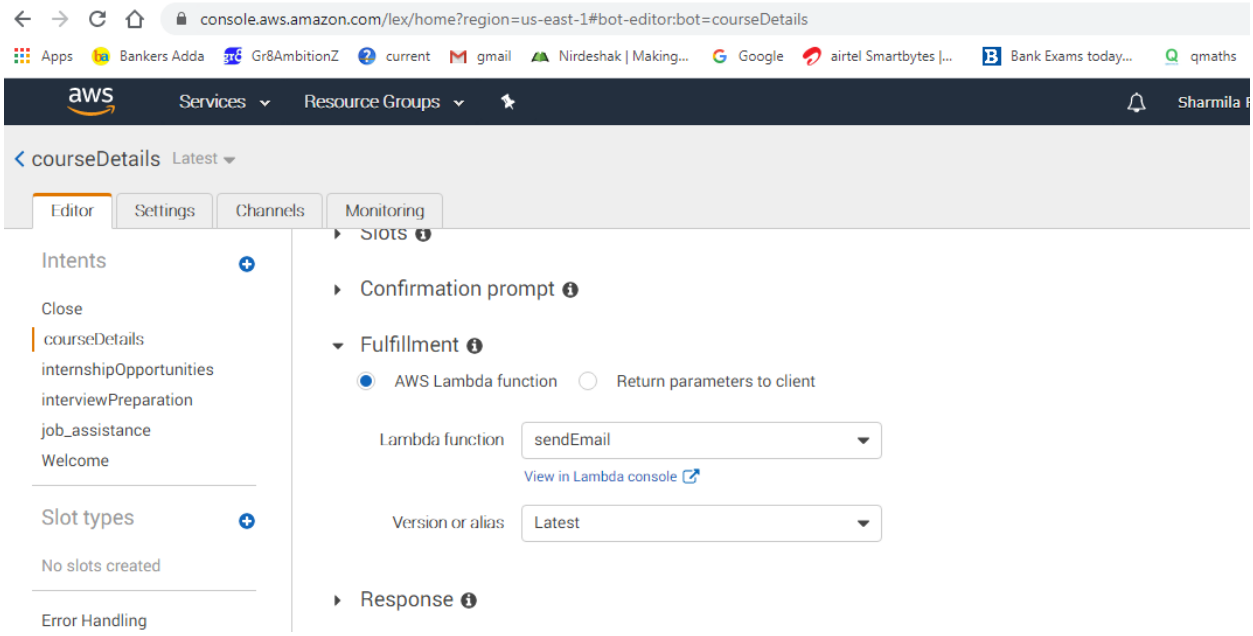
For courseDetails intent add the following parameters:

Sample utterances



Fullfillment

- Select AWS Lammda Function and choose sendEmail function.

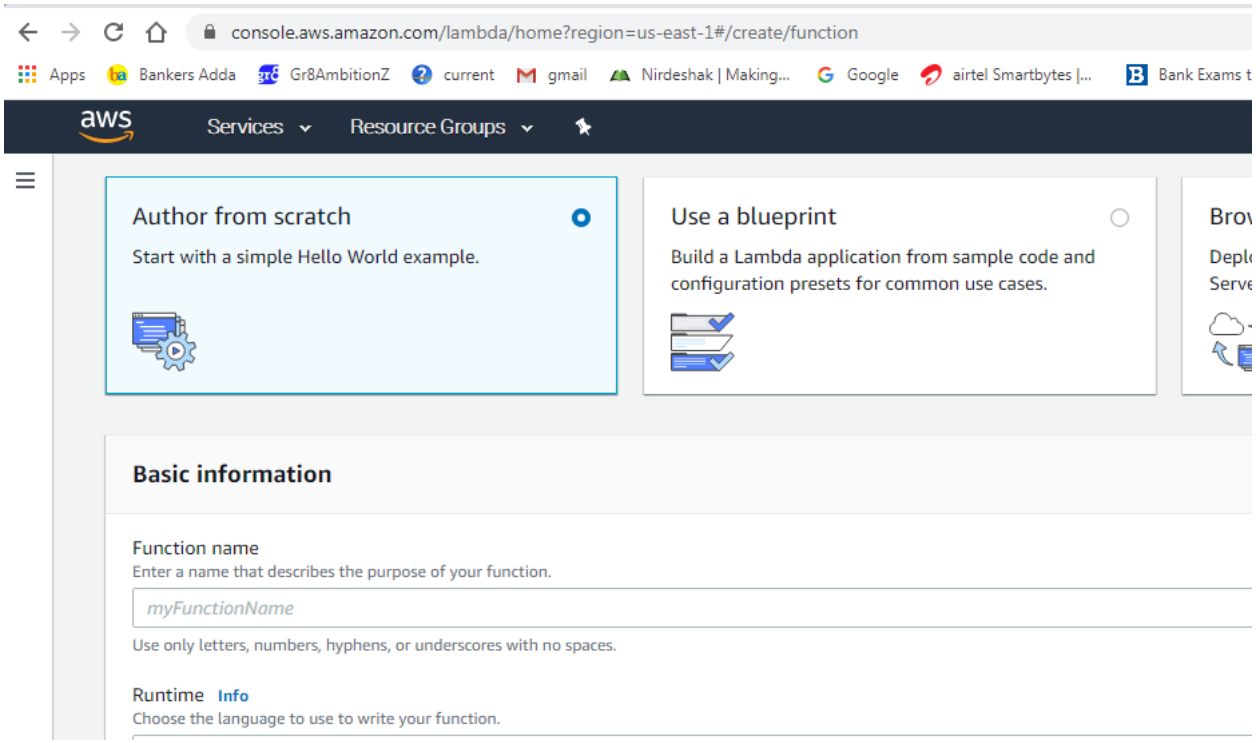


To create the Lambda function (console)

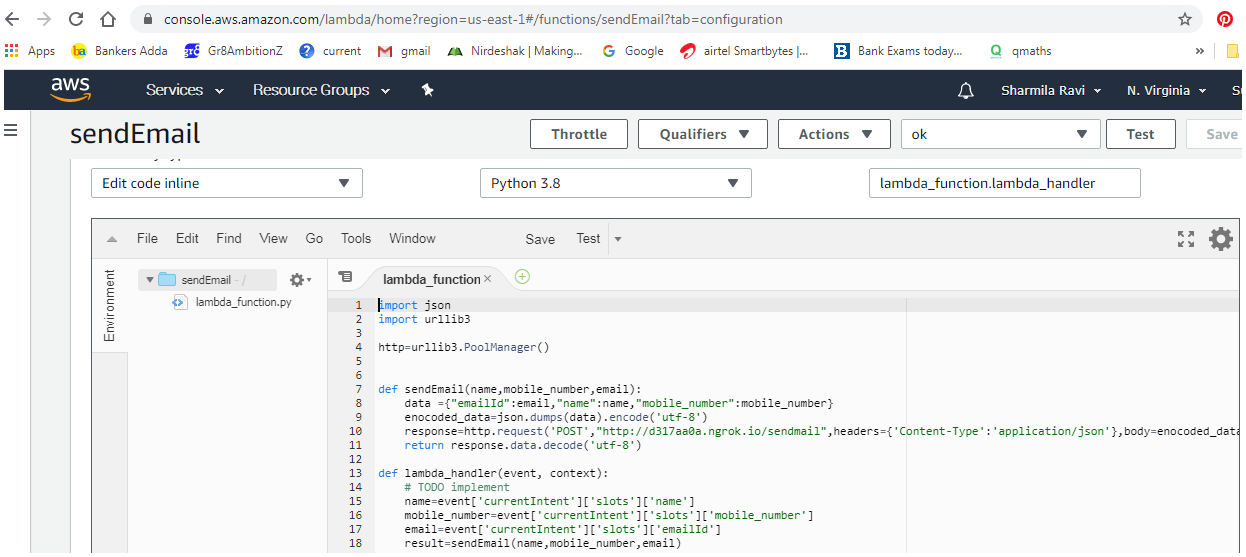
- Sign in to the AWS Management Console and open the AWS Lambda console at https://console.aws.amazon.com/lambda/.
- Choose Create function.
- On the Create function page, choose Use a Author from scratch
- On the Basic information page, do the following
- Type a Lambda function name (sendEmail)
- For runtime choose python
- Leave the other default values.
- Choose create function.

Test the Lambda function.

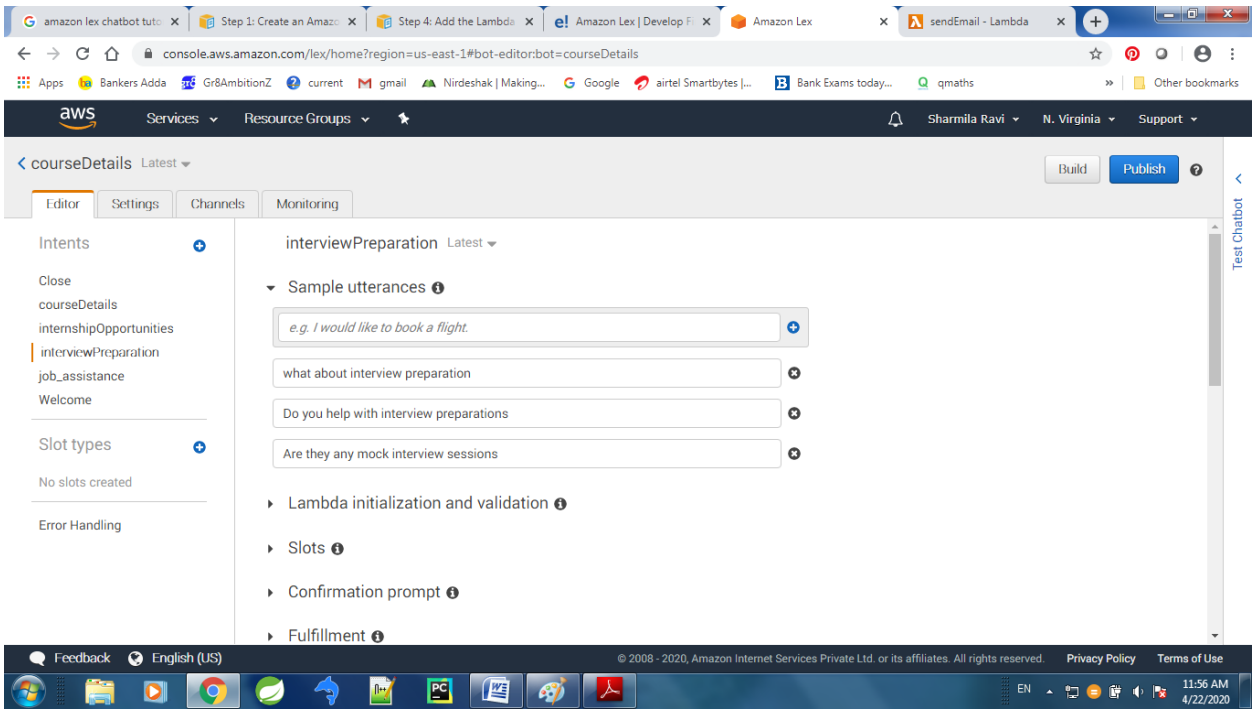
- Choose Select a test event, Configure test events



Add the below code in the function code section for the lamda funtion.



For interviewPreparation intent add the following values

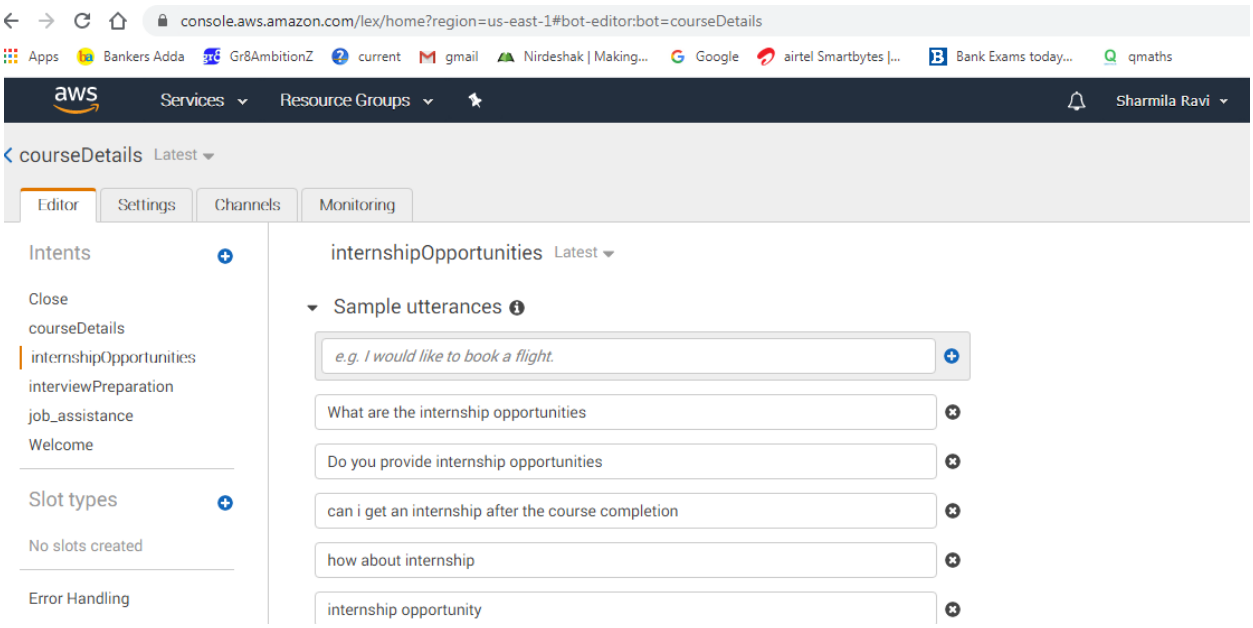


Responses:

- We'll have at least 3 Mock interviews for each candidate to prepare them for a job interview.

For internshipOpportunities add the following paramaters:

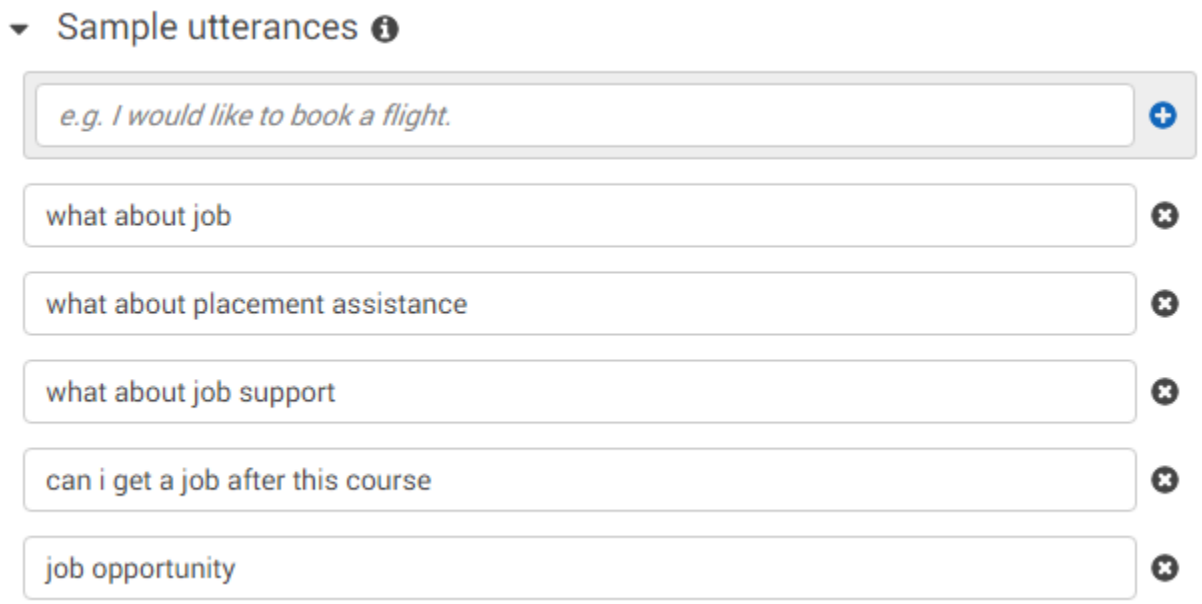
Sample utterances:



Response

- Both paid and unpaid internships are available with us.The selection for the internship will solely depend on your interview result.

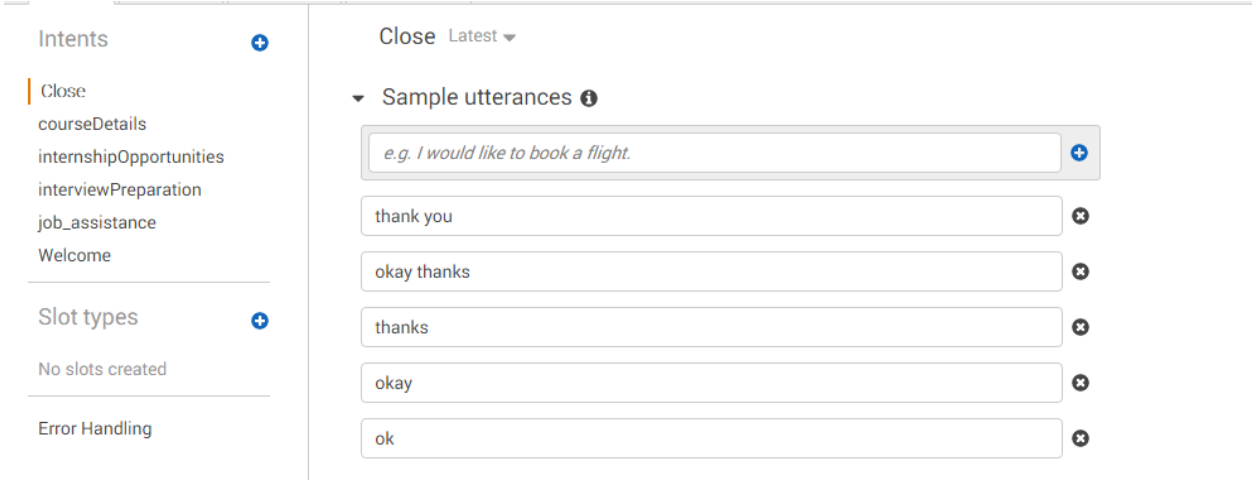
For job_assistance add the following parameters:



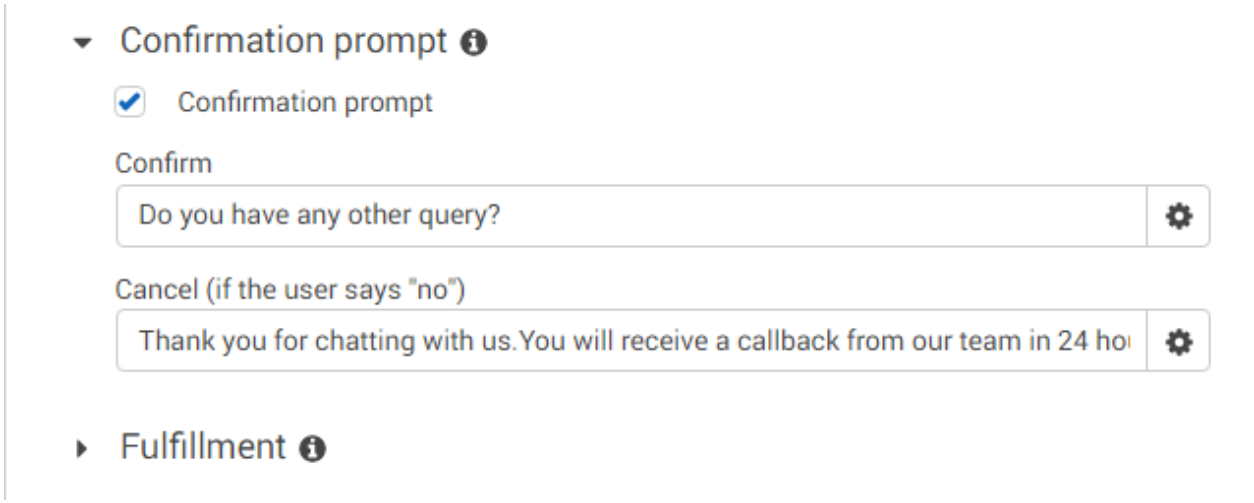
Responses:

- We'll provide referral to the candidates.We have our own custom portal where you can search for jobs.The link is:https://jobs.ineuron.ai

For close intent add the following parameters:



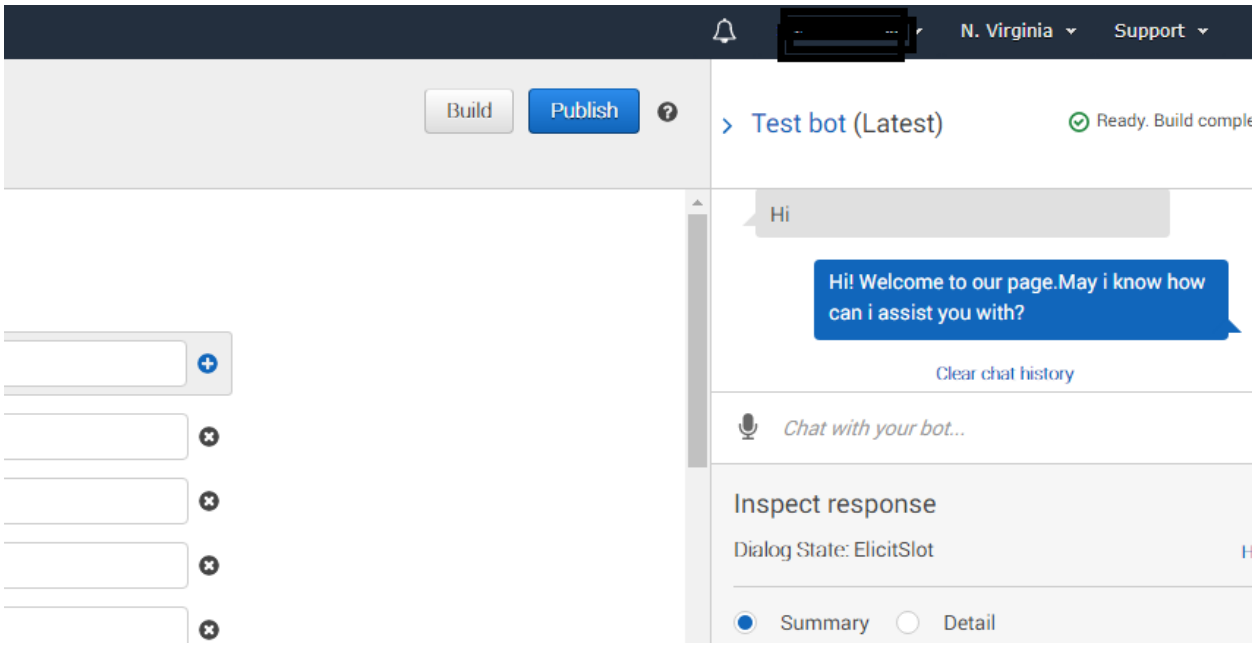
Confirmation Prompt



To build and test the bot

- To build the courseDetails bot, choose **Build**.
- Amazon Lex builds a machine learning model for the bot. When you test the bot, the console uses the runtime API to send the user input back to Amazon Lex. Amazon Lex then uses the machine learning model to interpret the user input.

To test the bot, in the **Test Bot** window, start communicating with your Amazon Lex bot.

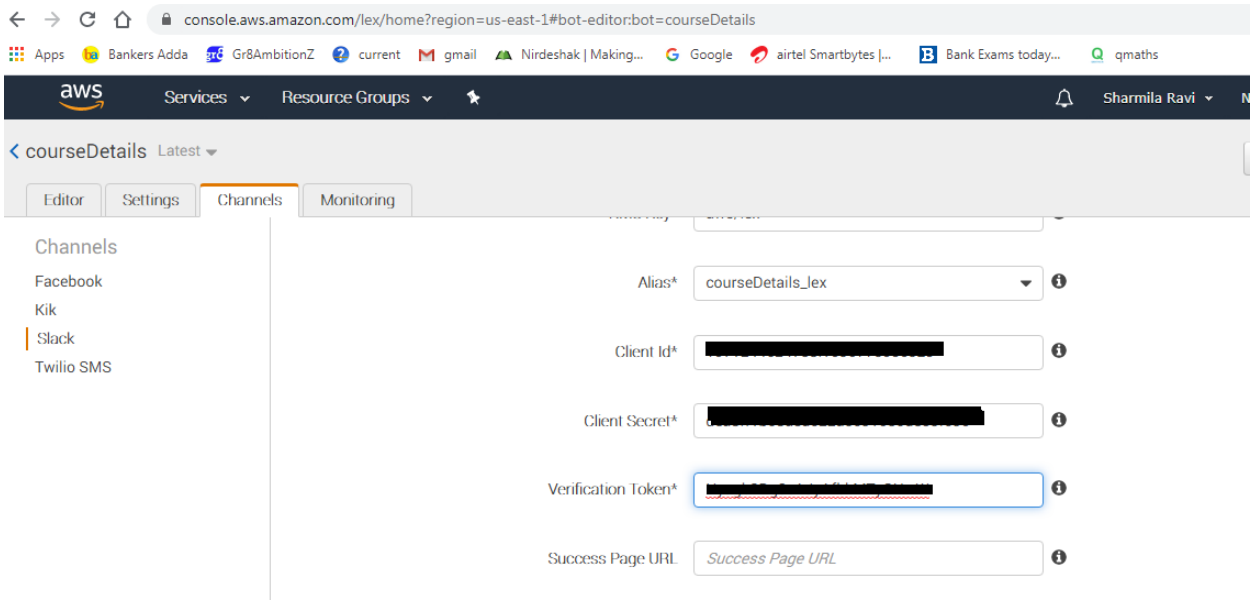


Integratin g Chatbot with slack

1. Sign in to the Slack API Console at http://api.slack.com .
2. Create an application.
3. After you have successfully created the application, Slack displays the Basic Information page for the application.
4. Configure the application features as follows:
5. In the left menu, choose Interactive Components.
6. Choose the toggle to turn interactive components on.
7. In the Request URL box, specify any valid URL. For example, you can use https://slack.com.
8. Choose Save Changes.
9. In the left menu, in Settings, choose Basic Information. Record the following application credentials:
    - **Client ID**
    - **Client Secret**
    - **Verification Token**

To integrate the Slack application with your Amazon Lex bot

- Sign in to the AWS Management Console, and open the Amazon Lex console at https://console.aws.amazon.com/lex/.
- Choose the Amazon Lex bot that you created.
- Choose the Channels tab.
- In the left menu, choose Slack.
- On the Slack page, provide the following:
    - Type a name. For example, `courseDetails`.
    - Choose "aws/lex" from the KMS key drop-down.
    - For Alias, choose the bot alias.
    - Type the Client Id, Client secret, and Verification Token, which you recorded in the preceding step. These are the credentials of the Slack application.
- Choose Activate.

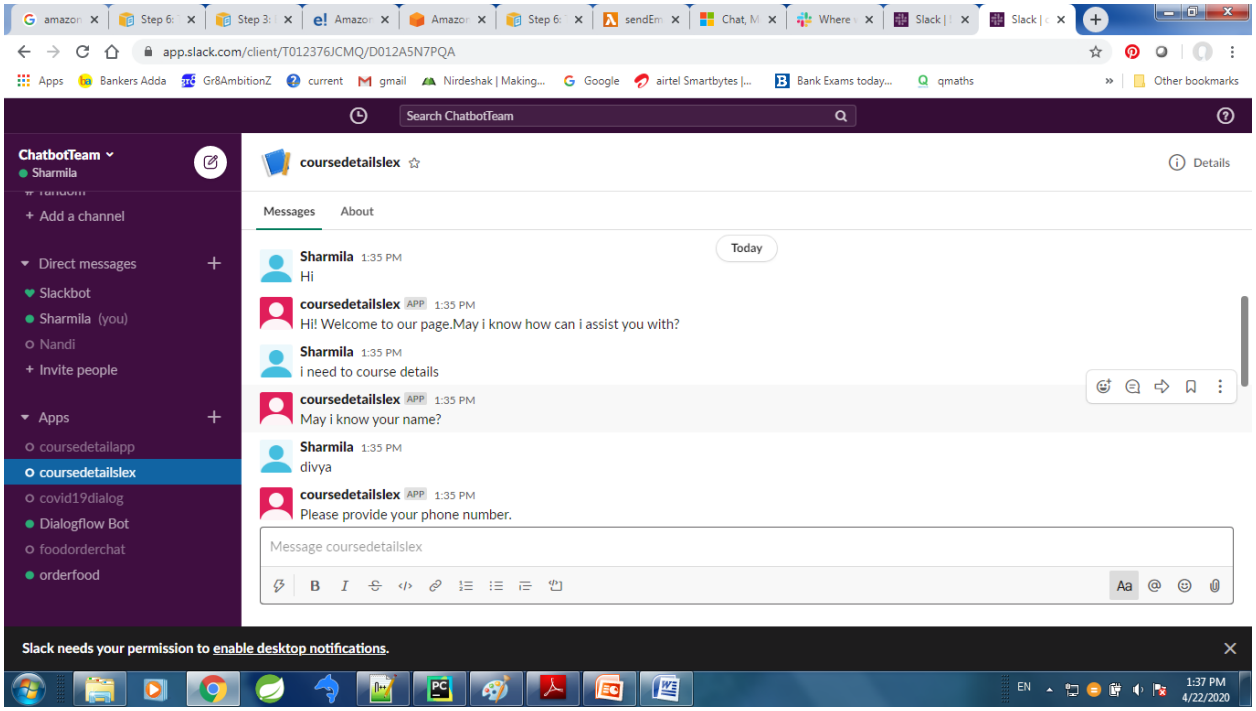


The console creates the bot channel association and returns two URLs (Postback URL and OAuth URL). Record them. In the next section, you update your Slack application configuration to use these endpoints as follows:

- The Postback URL is the Amazon Lex bot's endpoint that listens to Slack events. You use this URL:
    - As the request URL in the **Event Subscriptions** feature of the Slack application.
    - To replace the placeholder value for the request URL in the **Interactive Messages** feature of the Slack application.
- The OAuth URL is your Amazon Lex bot's endpoint for an OAuth handshake with Slack.

- Update the **OAuth & Permissions** feature as follows:
    - In the left menu, choose **OAuth & Permissions**.
    - In the **Redirect URLs** section, add the OAuth URL that Amazon Lex provided in the preceding step. Choose **Add a new Redirect URL**, and then choose **Save URLs**.
    - In the **Bot Token Scopes** section, choose two permissions in the **Select Permission Scopes** drop-down. Filter the list with the following text:
        - `chat:write`
        - `team:read`
- Choose **Save Changes**.
- Update the **Interactive Components** feature by updating the **Request URL** value to the Postback URL that Amazon Lex provided in the preceding step. Enter the postback URL that you saved in step 4, and then choose **Save Changes**.
- Subscribe to the **Event Subscriptions** feature as follows:
    - Enable events by choosing the **On** option.
    - Set the **Request URL** value to the Postback URL that Amazon Lex provided in the preceding step.
    - In the **Subscribe to Bot Events** section, subscribe to the `message.im` bot event to enable direct messaging between the end user and the Slack bot.
    - Save the changes.

1. Choose **Manage Distribution** under **Settings**. Choose **Add to Slack** to install the application. Authorize the bot to respond to messages.
2. You are redirected to your Slack team. In the left menu, in the **Direct Messages** section, choose your bot. If you don't see your bot, choose the plus icon (+) next to **Direct Messages** to search for it.
3. Engage in a chat with your Slack application, which is linked to the Amazon Lex bot. Your bot now responds to messages.

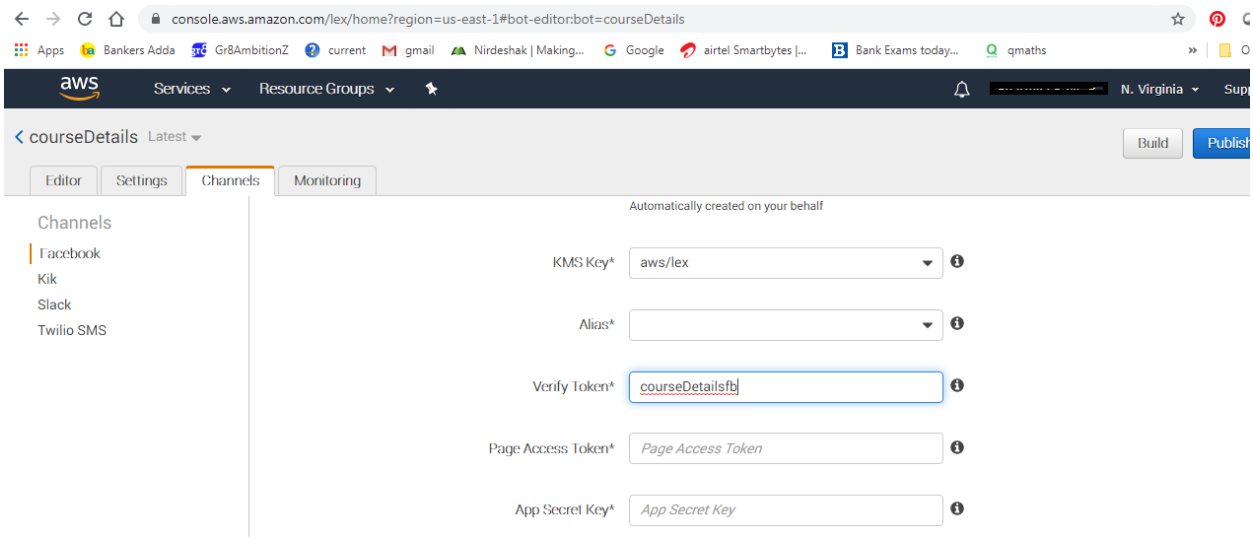


Integrating Chatbot with facebook

- Choose **Channels**.
- Choose **Facebook** under **Chatbots**. The console displays the Facebook integration page.
- On the Facebook integration page, do the following:
    - Type the following name: `courseDetails_fb`.
    - For **KMS key**, choose **aws/lex** .
    - For **Alias**, choose the bot alias.
    - For **Verify token**, type a token. This can be any string you choose (for example, `ExampleToken`). You use this token later in the Facebook developer portal when you set up the webhook.
- Choose **Activate**.
- The console creates the bot channel association and returns a callback URL. Write down this URL.
- On the Facebook developer portal, choose your app.
- Choose the **Messenger** product, and choose **Setup webhooks** in the **Webhooks** section of the page.

On the **webhook** page of the subscription wizard, do the following:

- For **Callback URL**, type the callback URL provided in the Amazon Lex console earlier in the procedure.



- For **Verify Token**, type the same token that you used in Amazon Lex.
- Choose **Subscription Fields** (**messages**, **messaging_postbacks**, and **messaging_optins**).
- Choose **Verify and Save**. This initiates a handshake between Facebook and Amazon Lex.
- Open your Facebook page, and choose **Message**